

# RELATÓRIO TÉCNICO

## DESENVOLVIMENTO WEB PORTFÓLIO PESSOAL



**ELABORADO POR:** Geremy Santos

**FORMADOR:** Willian Pires

DATA: 20/06/2024

# **Relatório Técnico**

## **Introdução**

Este relatório técnico destina-se a fornecer uma visão geral e uma análise do portfólio online criado por Jeremy Santos, no âmbito da disciplina de Desenvolvimento Web.

O presente relatório também tem como objetivo relatar o processo de criação de um portfólio pessoal utilizando as tecnologias **HTML**, **CSS** e **JavaScript**. Neste contexto, o desenvolvimento deste projeto permitiu a aplicação prática de conceitos essenciais de design e programação web, bem como a oportunidade de demonstrar de forma criativa e profissional as habilidades e experiências do autor. Ao longo deste relatório, serão abordados os detalhes do planejamento, implementação e aprimoramentos realizados no portfólio, destacando as decisões tomadas e os desafios enfrentados durante o processo de desenvolvimento.

## Desenvolvimento

### Funcionalidades e Interatividade do Portfólio

O portfólio apresenta diversas funcionalidades interativas que contribuem para uma experiência do usuário envolvente e dinâmica. Vamos destacar algumas das principais funcionalidades e como foram implementadas no código HTML, CSS, e JavaScript:

**Navegação:** A navegação do portfólio é facilitada pela presença de âncoras que direcionam o usuário para diferentes seções da página. Isso é alcançado por meio da atribuição de IDs específicos a cada seção e a utilização de links ancorados para permitir a navegação suave pelo conteúdo.

```
<ul class="nav_links" id="nav-links">
  <li><a href="#home">Home</a></li>
  <li><a href="#about">About</a></li>
  <li><a href="#service">Service</a></li>
  <li><a href="#portfolio">Projects</a></li>
  <li><a href="#contact" class="btn">Contact</a></li>
</ul>
```

**Animações:** Um ficheiro JavaScript foi incorporada ao código HTML para adicionar efeitos de animação durante o scroll da página. Isso cria um impacto visual agradável, revelando elementos conforme o usuário rola a página, tornando a experiência mais dinâmica e atrativa.

**Formulário de Contato:** O formulário de contato permite que os visitantes do portfólio enviem mensagens diretamente para o dono do portfolio. Os campos de entrada obrigatórios foram especificados com o atributo "required" para garantir que o usuário preencha as informações necessárias antes de enviar. Além disso, um botão "ENVIAR" foi implementado com a tag <input> de tipo "submit" para acionar o envio do formulário:

```
<input class="btn" type="submit" value="ENVIAR">
```

**Ícones de Redes Sociais:** Para promover a interação e conexão nas redes sociais, ícones correspondentes a plataformas como Twitter, Facebook, Instagram e LinkedIn foram incluídos no portfólio. Cada ícone é vinculado à respectiva página de Geremy nessas redes sociais, permitindo que os usuários cliquem e se conectem facilmente.

```
<div class="contact_socials">  
  <a href="#"><i class="ri-twitter-x-fill"></i></a>  
  <a href="#"><i class="ri-facebook-fill"></i></a>  
  <a href="#"><i class="ri-instagram-line"></i></a>  
  <a href="#"><i class="ri-linkedin-box-fill"></i></a>  
</div>
```

Essas funcionalidades interativas foram cuidadosamente integradas ao código HTML do portfólio, para melhorar a experiência do usuário. A combinação desses elementos cria um portfólio dinâmico e eficaz que reflete positivamente as habilidades de design de interfaces.

### **Design Responsivo no Portfólio**

O portfólio foi desenvolvido com um design responsivo, o que significa que o layout e os elementos da página se adaptam de forma fluida a diferentes dispositivos e tamanhos de tela, proporcionando uma experiência consistente e otimizada para todos os usuários.

Abaixo estão algumas das técnicas e práticas implementadas para garantir a responsividade do portfólio:

**Media Queries:** Foram utilizadas media queries no código CSS do portfólio para definir estilos específicos para diferentes tamanhos de tela. Isso inclui ajustar o tamanho e a disposição dos elementos, as margens e os espaçamentos, as fontes e outros aspectos visuais para garantir uma aparência bem apresentada em dispositivos pequenos, médios e grandes.

**Layout Flexível:** O layout do portfólio foi estruturado com unidades flexíveis, como porcentagens (%) e root ems (rem), em vez de valores fixos em pixels (px), para permitir que os elementos se redimensionem proporcionalmente conforme o tamanho da tela é alterado. Isso garante que o conteúdo se ajuste dinamicamente sem distorções ou perda de funcionalidade em diferentes dispositivos.

```
Exemplo: width: 100%;  
padding: 2rem;
```

**Imagens Responsivas:** Imagens no portfólio foram configuradas com atributos `srcset` e `sizes` para fornecer versões de resolução apropriadas com base no dispositivo do usuário. Isso ajuda a reduzir o tempo de carregamento da página em dispositivos móveis e garante que as imagens sejam exibidas com qualidade e clareza ideais em todas as resoluções de tela.

```

```

**Grid Layouts:** A utilização de sistemas de grid, como Flexbox ou CSS Grid, no código CSS auxilia na criação de layouts mais flexíveis e adaptáveis. Essas técnicas permitem organizar e reorganizar automaticamente os elementos da página em diferentes tamanhos de tela, mantendo a estrutura e a legibilidade do conteúdo.

```
.header_container{  
padding-top: 8rem;  
display: grid;  
gap: 2rem;  
isolation: isolate;  
}
```

Através dessas práticas e técnicas de design responsivo implementadas no código HTML e CSS do portfólio, o site se adapta de forma eficaz a uma variedade de dispositivos e resoluções de tela, proporcionando uma experiência de usuário agradável e consistente independentemente do meio de acesso.

## Ficheiro HTML

O código HTML apresentado nesta documentação representa a estrutura e o conteúdo do portfólio digital.

### Visão Geral do Código HTML:

O código HTML do portfólio de é estruturado de forma a fornecer uma experiência de usuário atraente e funcional. A seguir, teremos uma visão geral das principais características e elementos utilizados no código:

**Estrutura Basica:** O código inicia com uma estrutura HTML básica, contendo as tags <html>, <head> e <body> para definir o documento HTML e seu conteúdo.

**Seções e Divs:** O conteúdo do portfólio é organizado em diversas seções utilizando as tags <section> e <div>, permitindo uma divisão clara dos diferentes aspectos, como feedbacks, formulário de contato e informações de rodapé.

**Elementos de Texto:** Diversos elementos de texto são utilizados, como <h2>, <h4>, <p>, <h2>, <i>, e <br>, para estruturar e apresentar o conteúdo de forma legível e atraente.

**Tabela de habilidades:** Esse trecho de código HTML representa uma tabela com informações sobre habilidades em diferentes tecnologias web, como HTML, CSS e JavaScript. Aqui está uma explicação do código:

```
<div id="tabela_itens">
  <table border="1px" id="tabela_abilidades">
    <tr>
      <th>HTML</th>
      <th>CSS</th>
      <th>JavaScript</th>
    </tr>
    <tr>
      <td>Avançando</td>
      <td>Avançado</td>
      <td>Intermediario</td>
    </tr>
  </table>
</div>
```

**<div id="tabela\_itens">** : Define uma divisão (ou seção) com o identificador (id) "tabela\_itens". Essa divisão pode ser estilizada ou manipulada por meio de CSS ou JavaScript.

**<table border="1px" id="tabela\_abilidades">** : Define uma tabela com borda simples de 1 pixel e o identificador "tabela\_abilidades". A tabela é a estrutura principal que conterá as linhas e colunas de dados.

**<tr>** : Define uma linha dentro da tabela.

**<th>HTML</th>, <th>CSS</th>, <th>JavaScript</th>** : Define células de cabeçalho na primeira linha da tabela. Neste caso, são os títulos das colunas da tabela que representam as habilidades em HTML, CSS e JavaScript.

**<tr>** : Define uma nova linha na tabela para inserir os dados das habilidades.

**<td>Avançando</td>, <td>Avançado</td>, <td>Intermediário</td>** : Define células de dados nas respectivas colunas da segunda linha da tabela. Aqui, cada célula contém a informação sobre o nível de habilidade na tecnologia correspondente.

**Formulário de Contato:** Um formulário de contato é incluído utilizando as tags

**<form>**, **<input>** e **<textarea>**, possibilitando aos visitantes do site enviar mensagens e fazer agendamentos.

```
<section class="formulario">
  <form action="">
    <h2>Make Your Appointment!</h2>
    <input type="name" required placeholder="Insira seu nome
completo:">
    <input type="email" class="form_text" required
placeholder="Insira seu e-mail:">
    <input type="number" class="form_text" required
placeholder="Seu telefone:">
    <textarea class="text" name="" required
placeholder="Deixe uma mensagem:"></textarea>
    <div class="btn-enviar">
      <input class="btn" type="submit" value="ENVIAR">
    </div>
  </form>
</section>
```

**Elementos Visuais:** Ícones e emojis são incorporados ao design usando as tags `<i>` e caracteres especiais, adicionando elementos visuais interessantes e comunicativos ao conteúdo:

```
<i class="ri-facebook-fill"></i>
<i class="ri-instagram-line"></i>
<i class="ri-linkedin-box-fill"></i>
```

**Estilização com CSS:** O HTML faz uso de estilos CSS para a formatação visual, layout e design do site. Classes e IDs foram atribuídos aos elementos HTML para aplicar estilos personalizados:

```
<link rel="stylesheet" href="styles.css">
```

**Scripts Externos:** Dois scripts são referenciados no final do código, para um arquivo JavaScript `main.js`, sugerindo funcionalidades adicionais como animações e interações dinâmicas:

```
<script src="main.js"></script>
```

A importância do código HTML está diretamente relacionada à forma como ele estrutura e apresenta o conteúdo do portfólio, influenciando a usabilidade, acessibilidade e experiência geral do usuário. Um código bem escrito e otimizado não só melhora a eficácia do site, mas também contribui para a visibilidade nos mecanismos de busca e a satisfação do usuário.

Em resumo, o código HTML deste projecto, utiliza uma estrutura bem organizada e elementos diversificados para apresentar de forma eficaz seu trabalho e informações. A integração de HTML com CSS e scripts externos contribui para a criação de um site dinâmico e visualmente atraente, adequado para exibir projetos de design de interface de forma impactante e profissional.



## Ficheiro CSS

Este relatório técnico também, tem como objetivo analisar e descrever a estrutura e funcionalidades do código CSS apresentado. O código em questão define variáveis personalizadas, estilos para elementos HTML e classes para diferentes seções do projeto. A utilização de variáveis personalizadas ajuda a manter a consistência visual e facilita a manutenção e a escalabilidade do código CSS. Além disso, são aplicados estilos para tornar o layout responsivo, garantindo uma experiência de usuário satisfatória em diferentes dispositivos. Ao longo deste relatório, serão detalhadas as diferentes partes do código, suas funcionalidades e como contribuem para a estilização e funcionalidade do projeto em questão.

### Objetivos do código CSS:

**Melhorar a Estilização do Portfolio:** O código CSS visa aprimorar a apresentação visual do Portfolio, aplicando estilos consistentes e atraentes para diferentes elementos, como textos, botões, formulários e layout em geral.

**Garantir Consistência Visual:** Um dos objetivos é assegurar que todos os elementos do site sigam uma identidade visual coesa, mantendo uma harmonia de cores, tipografia e espaçamento para transmitir uma mensagem unificada.

**Tornar o Layout Responsivo:** O código CSS busca tornar o layout do site responsivo, ou seja, capaz de se adaptar a diferentes tamanhos de tela, como smartphones, tablets e desktops, garantindo uma experiência de usuário positiva em qualquer dispositivo.

**Facilitar Manutenção e Escalabilidade:** A utilização de variáveis personalizadas e uma estrutura organizada no código CSS tem como objetivo facilitar a manutenção futura do projeto, permitindo alterações rápidas e escalabilidade conforme o site cresce e evolui.

**Contribuir para a Usabilidade:** O CSS visa não apenas melhorar a estética visual do site, mas também aprimorar a usabilidade, certificando-se de que os elementos sejam claramente legíveis, interativos e acessíveis para os usuários.

## Metodologia de Desenvolvimento do Código CSS:

As etapas de desenvolvimento envolveram a criação de estilos para diferentes elementos do site, como definições de cores, tipografia, espaçamento, alinhamento, bordas, entre outros. Foram utilizadas classes e seletores para aplicar estilos de forma seletiva e modular, facilitando a manutenção e a reutilização de estilos. Decisões de Design foram tomadas considerando a identidade visual do autor do projecto. As cores, fontes e espaçamentos foram escolhidos de forma a transmitir profissionalismo, clareza e modernidade.

## Estrutura do código CSS:

Primeiro comecemos por importar um estilo de fonte personalizado:

```
@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100..900;1,100..900&display=swap');
```

Essa linha de código em CSS está importando um estilo de fonte personalizado da Google Fonts chamado "Montserrat" para ser usado no seu projeto. A URL especificada está apontando para o arquivo CSS que contém as instruções para carregar e exibir a fonte Montserrat em diferentes pesos e estilos (normal e itálico) em seu site. A propriedade 'display=swap' é usada para garantir que o texto seja exibido com a fonte personalizada assim que estiver disponível, evitando flashes de texto não estilizado.

Em seguida, essas linhas de código em CSS estão definindo variáveis personalizadas, que podem ser reutilizadas em diferentes partes do seu estilo CSS para manter a consistência e facilitar alterações futuras.

```
:root {
  --primary-color: #feaa5c;
  --secondary-color: #e14418;
  --extra-ligth: #f3f4f6;
  --white: #ffffff;
  --gradient-1: linear-gradient(to bottom right, #151515, #000000);
  --gradient-2: linear-gradient(
    to bottom,
    var(--primary-color),
    var(--secondary-color)
  );
  --max-width: 1200px;
}
```

### Aqui está a explicação de cada elemento:

- **:root:** Esse seletor representa o elemento raiz do documento, geralmente usado para definir variáveis globais que podem ser acessadas em qualquer lugar do CSS.
- **--primary-color: #feaa5c;;** Define a variável --primary-color como a cor laranja especificada.
- **--secondary-color: #e14418;;** Define a variável --secondary-color como a cor vermelha especificada.
- **--extra-light: #f3f4f6;;** Define a variável --extra-light como a cor cinza claro especificada.
- **--white: #ffffff;** Define a variável --white como a cor branca.
- **--gradient-1: linear-gradient(to bottom right, #151515, #000000);;** Define a variável --gradient-1 como um gradiente linear que vai do preto (#151515) ao preto (#000000) na direção bottom right.
- **--gradient-2: linear-gradient(to bottom, var(--primary-color), var(--secondary-color));;** Define a variável --gradient-2 como um gradiente linear que vai da cor especificada em --primary-color para a cor especificada em --secondary-color na direção bottom (para baixo).
- **--max-width: 1200px;;** Define a variável --max-width com o valor de 1200 pixels, que pode ser usado para limitar a largura máxima de certos elementos na página.

### Estilos para a seção do cabeçalho (header):

**.header:** Esta classe define o estilo para a seção do cabeçalho do site, com uma cor de fundo escura, texto em branco, padding para espaçamento interno e alinhamento centralizado. Exemplo de uso no HTML: <header class="header">.

## Estilos para a navegação (navbar):

**.nav\_bar:** Define o estilo para a barra de navegação, utilizando flexbox para centralizar os itens de navegação. Exemplo de uso no CSS:

```
.nav_bar{
  padding: 1rem;
  display: flex;
  align-items: center;
  justify-content: space-between;
  gap: 2rem;
  background-image: var(--gradient-1);
}
```

**.nav\_bar a:** Estiliza os links dentro da navegação, definindo cor do texto, removendo sublinhado e aplicando margens entre os links:

```
.nav_bar a{
  font-size: 1.2rem;
  font-weight: 600;
  color: var(--white);
}
```

## Estilos para os botões (btn):

**.btn:** Define o estilo para os botões do site, com uma cor de fundo destacada, texto em branco, padding para espaçamento interno, bordas arredondadas e remoção de sublinhado:

```
.btn{
  padding: .75rem 1.5rem;
  font-size: 1rem;
  font-weight: 500;
  color: var(--white);
  background-image: var(--gradient-2);
  outline: none;
  border: none;
  border-radius: 5rem;
  cursor: pointer;
  box-shadow: 5px 5px 10px rgb(0, 0, 0, 0.2);
}
```

**.btn:hover:** Define o estilo para aumentar o tamanho dos botões do site, quando o cursor estiver em cima do botão, numa escala de 1.1, ou seja 100% do seu tamanho.

```
.btn:hover{
  transform: scale(1.1);
}
```

### Estilos para o rodapé (footer):

```
.footer{
  padding: 1rem;
  font-size: 0.9rem;
  color: var(--extra-ligth);
  text-align: center;
}
```

**.footer:** Estiliza o rodapé do site com uma cor de fundo escura, texto em branco, padding e alinhamento central. Exemplo de uso no HTML: `<footer class="footer">`.

Essas seções do código CSS estão organizadas de forma lógica e modular, facilitando a aplicação e manutenção dos estilos no site. Cada classe definida pode ser atribuída a elementos HTML correspondentes para garantir a consistência visual em todo o site.

Neste trecho de código CSS, está sendo utilizada uma regra de media query para definir estilos específicos quando a largura da tela for maior que 576 pixels. Ou seja, esses estilos serão aplicados apenas em dispositivos com uma tela mais larga.

```
@media (width > 576px){
  .service_grid{
    grid-template-columns: repeat(2, 1fr);
  }
  .portfolio_grid{
    grid-template-columns: repeat(2, 1fr);
  }
  .portfolio_banner{
    grid-template-columns: repeat(2, 1fr);
    text-align: left;
  }
}
```

Dentro dessa regra **@media (width > 576px)**, encontramos as seguintes definições de estilo:

- **.service\_grid:** Define que os elementos com a classe **.service\_grid** terão sua disposição em grid com duas colunas, onde cada coluna ocupará a mesma fração de espaço disponível.
- **.portfolio\_grid:** Define que os elementos com a classe **.portfolio\_grid** também terão sua disposição em grid com duas colunas, onde cada coluna ocupará a mesma fração de espaço disponível.

- **.portfolio\_banner:** Define que os elementos com a classe **.portfolio\_banner** terão sua disposição em grid com duas colunas, onde cada coluna irá ocupar a mesma fração de espaço disponível. Além disso, o texto dentro desses elementos terá alinhamento à esquerda.

Em resumo, esse trecho de código ajusta o layout das grades de serviços, portfólio e banners em dispositivos com largura de tela maior que 576 pixels para que tenham duas colunas, proporcionando um design responsivo e adaptado para telas maiores

### **Resultados Alcançados:**

O código CSS fornece estilos para diferentes seções do site, incluindo cabeçalho, navegação, botões e rodapé. Os estilos aplicados resultam em uma visualização coesa e agradável das diferentes partes do site. A organização modular do código facilita a manutenção e aplicação dos estilos em todo o projecto.

## Ficheiro JavaScript

Neste relatório técnico apresentaremos uma análise detalhada do código JavaScript fornecido, que faz parte da implementação e criação do Portfolio. O código em questão é responsável por controlar a interatividade do menu de navegação, a animação de abertura e fechamento do menu, além de aplicar efeitos de revelação ao rolar a página.

A análise abordará a eficiência, legibilidade e boas práticas presentes no código, com o intuito de avaliar sua qualidade e possíveis melhorias. Serão destacados pontos-chave como a utilização de eventos, manipulação do DOM, aplicação de classes dinâmicas e o uso de algumas ferramentas e técnicas para criar efeitos de animação.

O código JavaScript em questão foi desenvolvido com o objetivo de aprimorar a experiência de navegação em um site, mediante a implementação de funcionalidades interativas e efeitos visuais agradáveis, que serão apresentados a seguir.

**Controle do Menu de Navegação:** O código permite ao usuário interagir com o menu de navegação por meio de um botão. Ao clicar no ícone do menu (representado pela classe "ri-menu-3-line"), a lista de links de navegação é exibida ou ocultada, proporcionando ao usuário uma forma intuitiva de explorar o conteúdo do site.

```
menuBtn.addEventListener("click", (e) => {  
  navLinks.classList.toggle("open");  
});
```

**Animação do Ícone do Menu:** Durante a interação com o menu de navegação, o ícone do botão do menu é animado para refletir o estado atual do menu. Quando o menu é aberto, o ícone pode mudar para indicar essa ação ao usuário, facilitando a compreensão visual da operação realizada.

```
menuBtn.addEventListener("click", (e) => {  
  navLinks.classList.toggle("open");  
  
  const isOpen = navLinks.classList.contains("open");  
  menuBtnIcon.setAttribute(  
    "class",  
    isOpen ? "ri-close-line" : "ri-menu-3-line"  
  );  
});
```

**Efeitos de Revelação ao Rolar a Página:** O código utiliza efeitos de animação ao rolar a página. Esses efeitos proporcionam uma experiência de visualização dinâmica, revelando gradualmente os elementos conforme eles entram na área visível da tela. Isso contribui para uma navegação fluida e atraente, estimulando a interação do usuário com o conteúdo do site. Exemplos:

```
ScrollReveal().reveal(".header_content .section_description", {  
  ...scrollRevealOption,  
  delay: 500,  
});  
  
ScrollReveal().reveal(".header_content .header_btn", {  
  ...scrollRevealOption,  
  delay: 1000,  
});
```

**Organização em Seções:** O código está estruturado em seções que correspondem às diferentes partes do site, como cabeçalho, sobre, serviços e portfólio. Cada seção tem sua própria configuração de animação, permitindo uma personalização dos efeitos aplicados em cada área do site. Isso contribui para uma experiência mais coesa e envolvente para o usuário, direcionando sua atenção conforme navega pelas seções.

### **Objectivos da implementação do código JavaScript:**

**Melhorar a experiência de navegação:** O código busca proporcionar uma experiência de navegação mais intuitiva e agradável para os usuários, facilitando o acesso aos conteúdos do site por meio de um menu de navegação responsivo e interativo.

**Estimular a interação do usuário:** Ao implementar animações e efeitos visuais durante a navegação, o código busca estimular a interação dos usuários com o site, tornando a experiência mais envolvente e dinâmica.

**Facilitar o entendimento da estrutura do site:** A animação do ícone do menu e os efeitos de revelação ao rolar a página contribuem para tornar a estrutura do projecto mais clara e compreensível para os visitantes, direcionando sua atenção para os diferentes conteúdos apresentados.



**Personalizar a experiência de navegação por seção:** A organização do código em diferentes seções permite a personalização de animações e efeitos específicos para cada parte do site, criando uma experiência mais coesa e coesa para os usuários conforme eles exploram o conteúdo.

**Aprimorar a estética e a atratividade visual do Portfolio:** Os efeitos de animação e a revelação gradual dos elementos visuais contribuem para a estética do site, tornando-o mais atrativo e visualmente impactante para os visitantes.

Em resumo, o código tem como objetivos principais melhorar a experiência geral de navegação, incentivar a interação dos usuários, facilitar a compreensão da estrutura do site, personalizar a experiência por seção e aprimorar a estética visual do site. Esses objetivos visam proporcionar uma experiência de usuário positiva e eficaz, contribuindo para o sucesso e a eficácia do site como um todo.

## **Divisão do código em seções lógicas e descrição de cada parte:**

### **Inicialização e Configuração Inicial:**

```
const navLinks = document.getElementById("nav-links");  
const menuBtn = document.getElementById("menu-btn");  
const menuBtnIcon = menuBtn.querySelector("i");
```

### **Controle de Menu Responsivo:**

O código começa definindo variáveis para acessar o elemento **nav-links**, o botão do menu **menu-btn** e o ícone dentro do botão do menu.

Em seguida, um evento de clique é adicionado ao **menu-btn**. Quando o botão é clicado, a classe **open** é alternada no elemento **nav-links**, o que abre ou fecha o menu.

Também é verificado se a classe **open** está presente no **nav-links**, e com base nisso, o ícone do botão do menu é alterado para exibir um ícone de fechar ou de menu.

**Função:** Esta seção do código é responsável por inicializar variáveis, configurar eventos iniciais e carregar quaisquer recursos necessários para o funcionamento da página. Aqui também são feitas as importações de bibliotecas e configurações iniciais.

## Manipulação de Eventos:

```
navLinks.addEventListener("click", (e) => {
  navLinks.classList.remove("open");
  menuBtnIcon.setAttribute("class", "ri-menu-3-line");
});

const scrollRevealOption = {
  distance: "50px",
  origin: "bottom",
  duration: 1000,
};
```

## Animações com ScrollReveal:

Variável `scrollRevealOption` é definida com algumas opções de animação, como distância, origem e duração.

**Função:** Esta parte do código trata da captura e manipulação de eventos de interface do usuário, como cliques de botão, scrolls ou interações com formulários. Aqui, são definidas as ações a serem executadas em resposta a esses eventos.

## Animações e Efeitos Visuais:

Várias chamadas do `ScrollReveal` são feitas para diferentes elementos na página, como cabeçalho, conteúdo sobre, serviços e portfólio:

```
//header container
ScrollReveal().reveal(".header_content h1", {
  ...scrollRevealOption,
});

ScrollReveal().reveal(".header_content .section_description", {
  ...scrollRevealOption,
  delay: 500,
});
```

```
ScrollReveal().reveal(".header_content .header_btn", {
  ...scrollRevealOption,
  delay: 1000,
});
```

```
//about container
ScrollReveal().reveal(".about_content .section_header", {
  ...scrollRevealOption,
});
```

```
ScrollReveal().reveal(".about_content .section_description", {
  ...scrollRevealOption,
  delay: 500,
});
```

```
ScrollReveal().reveal(".about_content .about_btn", {  
  ...scrollRevealOption,  
  delay: 1000,  
});
```

```
//service container  
ScrollReveal().reveal(".service_card", {  
  ...scrollRevealOption,  
  interval: 500,  
});
```

```
//portfolio container  
ScrollReveal().reveal(".portfolio_card",{  
  duration:1000,  
  interval:500,  
});
```

Para cada elemento selecionado, a animação é aplicada com base nas opções definidas em scrollRevealOption, e algumas animações têm atrasos específicos para criar uma transição suave.

**Função:** Nesta seção, são definidas as animações e efeitos visuais a serem aplicados aos elementos da página. Isso pode incluir transições suaves, transformações de CSS e efeitos de scroll.

### Navegação e Interatividade:

Assim como já foi especificado mais acima, este código JavaScript controla a abertura/fechamento de um menu responsivo ao clicar em um botão, e utiliza a biblioteca ScrollReveal para adicionar animações conforme o usuário rola a página, revelando de forma elegante e progressiva certos elementos da página conforme são visualizados.

**Função:** Aqui são definidas as funcionalidades de interatividade, como a navegação por menus, abas e seções deslizantes. Esta seção garante uma experiência de navegação intuitiva e amigável.

## **Conclusão**

Este relatório técnico tem como foco principal relatar o processo de aprendizado e criação dos códigos para o portfólio online desenvolvido por Jeremy Santos, no contexto da disciplina de Desenvolvimento Web. O objetivo é fornecer uma análise detalhada das etapas de planejamento, implementação e aprimoramento do projeto, destacando as decisões tomadas e os desafios enfrentados durante o desenvolvimento das funcionalidades interativas utilizando HTML, CSS e JavaScript.

A criação do portfólio online representou uma oportunidade valiosa de aplicar os conhecimentos teóricos adquiridos no curso na prática. Durante a implementação das funcionalidades interativas, foi necessário compreender e empregar os princípios fundamentais de cada linguagem de programação.

No código HTML, foram estruturados os elementos de forma organizada, garantindo uma base sólida para a construção do layout do portfólio. A marcação adequada do conteúdo e a utilização de tags semânticas contribuíram para a acessibilidade e indexação do site pelos motores de busca.

No CSS, a estilização e o design visual foram cuidadosamente desenvolvidos para proporcionar uma experiência estética agradável aos usuários. A utilização de seletores, propriedades e valores adequados foram essenciais para a formatação e o posicionamento dos elementos na página.

Já no JavaScript, as funcionalidades interativas foram implementadas para tornar o portfólio dinâmico e envolvente. A manipulação do DOM e o uso de eventos foram explorados para criar animações, transições e respostas em tempo real às interações do usuário.

Durante o processo de desenvolvimento, surgiram desafios que demandaram resolução criativa e aprofundamento dos conhecimentos técnicos. A identificação e correção de bugs, a otimização do desempenho e a garantia da compatibilidade entre os diferentes navegadores foram aspectos críticos abordados com atenção e dedicação. A aplicação prática dos conceitos de HTML, CSS e JavaScript permitiu não apenas a construção de um projeto funcional, mas também o desenvolvimento e aprimoramento das habilidades técnicas e criativas do autor.

O impacto dessa experiência não se limita apenas ao resultado final do portfólio, mas se estende ao conhecimento adquirido, às habilidades aprimoradas pela programação.

## Fontes e Referencias:

### Referências para o código HTML:

Youtube video: <https://youtu.be/8AmATH3vtAA>

Site W3school: <https://www.w3schools.com/html/default.asp>

Downloads dos ícones: <https://remixicon.com/>

### Referências para o código CSS:

Youtube video: <https://youtu.be/8AmATH3vtAA>

Site W3school: <https://www.w3schools.com/css/default.asp>

MDN Web Docs: guia completo de referência para HTML, CSS e JavaScript:  
<https://developer.mozilla.org/pt-BR/>

Curso online "CSS Avançado" na plataforma Udemy: <https://www.udemy.com/>

### Referências para o código JavaScript:

Youtube video: <https://youtu.be/8AmATH3vtAA>

Site W3school: <https://www.w3schools.com/js/default.asp>

Documentação Oficial do JavaScript: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>

**Elaborado por:** Jeremy santos